

PART THREE

**Qualifying the
Inefficiency of Equilibria**

CHAPTER 17

Introduction to the Inefficiency of Equilibria

Tim Roughgarden and Éva Tardos

Abstract

This chapter presents motivation and definitions for quantifying the inefficiency of equilibria in noncooperative games. We illustrate the basic concepts in four fundamental network models, which are studied in depth in subsequent chapters. We also discuss how measures of the inefficiency of equilibria can guide mechanism and network design.

17.1 Introduction

17.1.1 The Inefficiency of Equilibria

The previous two parts of this book provided numerous examples demonstrating that the outcome of rational behavior by self-interested players can be inferior to a centrally designed outcome. This part of the book is devoted to the question: *by how much?*

To begin, recall the Prisoner's Dilemma (Example 17.1). Both players suffer a cost of 4 in the unique Nash equilibrium of this game, while both could incur a cost of 2 by coordinating. There are several ways to formalize the fact that the Nash equilibrium in the Prisoner's Dilemma is inefficient. A qualitative observation is that the equilibrium is strictly Pareto inefficient, in the sense that there is another outcome in which all of the players achieve a smaller cost. This qualitative perspective is particularly appropriate in applications where the "cost" or "payoff" to a player is an abstract quantity that only expresses the player's preferences between different outcomes. However, payoffs and costs have concrete interpretations in many applications, such as money or the delay incurred in a network. We can proceed more quantitatively in such applications and posit a specific *objective function*, defined on the outcomes of the game, that numerically expresses the "social good" or "social cost" of an outcome. Two prominent objective functions are the *utilitarian* and *egalitarian* functions, defined as the sum of the players' costs and the maximum player cost, respectively. The Nash equilibrium in the Prisoner's Dilemma does not minimize either of these objective functions.

Introducing an objective function enables us to *quantify* the inefficiency of equilibria, and in particular to deem certain outcomes of a game *optimal* or *approximately optimal*. The primary goal of this part of the book is to understand when, and in what senses, game-theoretic equilibria are guaranteed to approximately optimize natural objective functions. Such a guarantee implies that selfish behavior does not have severe consequences, and thus the benefit of imposing additional control over players' actions is relatively small. Guarantees of this sort are particularly useful in many computer science applications, especially those involving the Internet, where implementing an optimal solution can be impossible or prohibitively expensive.

In the remainder of this section, we discuss different measures that quantify the inefficiency of equilibria. In Section 17.2, we illustrate these concepts and motivate Chapters 18–21 via several examples in network games. Section 17.3 demonstrates how these same concepts provide a comparative framework for mechanism and network design. Section 17.4 concludes with bibliographic notes and suggestions for further reading.

17.1.2 Measures of Inefficiency

Several measures of “the inefficiency of the equilibria of a game” have been considered. All of these measures are defined, roughly, as the ratio between the objective function value of an equilibrium of the game and that of an optimal outcome. To specify such a measure precisely, we must answer the following basic modeling questions.

- (1) How are the payoffs or costs of players expressed?
- (2) What objective function do we use to compare different outcomes of the game?
- (3) What is our definition of “approximately optimal”?
- (4) What is our definition of an “equilibrium”?
- (5) When there are multiple equilibria, which one do we consider?

We next discuss, at a high level, the most commonly studied answers to all of these questions. We give several examples in Section 17.2.

The answer to the first question will be some concrete payoff that players seek to maximize (such as money earned), or a cost that players aim to minimize (such as network delay). Both cases arise naturally in the applications studied in this book.

Second, we focus primarily on the utilitarian objective function, where the goal is to maximize the sum of players' payoffs or minimize the sum of players' costs. However, we also study the egalitarian objective function in Section 17.2.3 and Chapter 20. We call an outcome of a game *optimal* if it optimizes the chosen objective function. For example, in the Prisoner's Dilemma, the coordinated outcome is optimal for both the utilitarian and egalitarian objective functions. While in principle the measures of inefficiency below make sense for most objective functions, we can only expect the outcome of selfish behavior to approximate an optimal outcome when the objective function is related to the players' objectives.

Third, we quantify the extent to which a given outcome approximates an optimal one according to the ratio between the objective function values of the two outcomes. We consider only nonnegative objective functions, so this ratio is always nonnegative. (By convention, we interpret the ratio $c/0$ as 1 if $c = 0$ and as $+\infty$ if $c > 0$.) This ratio

is at least 1 for minimization objectives and at most 1 for maximization objectives. In either case, a value close to 1 indicates that the given outcome is approximately optimal. For example, in the Prisoner's Dilemma, the sum of the players' costs in the Nash equilibrium is 8; since the minimum-possible sum of costs is 4, the corresponding ratio for the equilibrium outcome is 2. As Section 17.4 discusses, this use of a ratio is directly inspired by many similar notions of approximation that have been widely studied in theoretical computer science. While other notions of approximation are possible, almost all work on quantifying the inefficiency of equilibria has followed the approach taken here.

Several equilibrium concepts have been studied in different applications. In this chapter, we confine our attention to Nash equilibria and their analogues in games where the set of players or strategies is not finite. One particularly important issue not addressed in this chapter is the relationship between the inefficiency of equilibria and the ability of players to *reach* an equilibrium. In other words, a bound on the inefficiency of the equilibria of a game is much more compelling if we expect players to learn or converge to one of these equilibria. In many of the games discussed in this part of the book, relatively weak assumptions imply that local, uncoordinated optimization by players leads to an equilibrium outcome in a reasonable amount of time (see Sections 4.7 and 29.3). Some important classes of network games, however, do not admit such convergence results. This fact motivated researchers to define novel notions of "equilibrium outcomes," which include all Nash equilibria but also allow players to wander among a set of unstable outcomes. In some applications, all such equilibria, and not just the Nash equilibria, are guaranteed to be approximately optimal. Chapter 19 briefly discusses some results of this type. See Section 17.4 for further details.

Finally, given a choice of an objective function and an equilibrium concept, a game may have different equilibria with different objective function values; recall, for example, the coordination games of Section 17.1.3. In such games, it is not clear which equilibrium should be compared to an optimal outcome. Section 17.1.3 discusses the two most popular approaches.

17.1.3 The Price of Anarchy and the Price of Stability

The *price of anarchy*, the most popular measure of the inefficiency of equilibria, resolves the issue of multiple equilibria by adopting a worst-case approach. Precisely, the price of anarchy of a game is defined as the ratio between the worst objective function value of an equilibrium of the game and that of an optimal outcome. Note that the price of anarchy of a game is defined with respect to a choice of objective function and a choice of equilibrium concept. For example, as shown in Section 17.2.3 below, the price of anarchy of a game is generally different for different choices of an objective function.

We are interested in identifying games in which the price of anarchy is close to 1; in these games, all equilibria are good approximations of an optimal outcome. We view selfish behavior as benign in such games. Put differently, the benefit provided by (possibly costly or infeasible) dictatorial control over the players' actions is reasonably small.

A game with multiple equilibria has a large price of anarchy even if only one of its equilibria is highly inefficient. The *price of stability* is a measure of inefficiency designed to differentiate between games in which *all* equilibria are inefficient and those in which *some* equilibrium is inefficient. Formally, the price of stability of a game is the ratio between the *best* objective function value of one of its equilibria and that of an optimal outcome. Of course, in a game with a unique equilibrium, its price of anarchy and price of stability are identical. For a game with multiple equilibria, its price of stability is at least as close to 1 as its price of anarchy, and it can be much closer (see Example 17.2 below).

A bound on the price of stability, which ensures only that one equilibrium is approximately optimal, provides a significantly weaker guarantee than a bound on the price of anarchy. Nevertheless, there are two reasons to study the price of stability. First, in some applications, a nontrivial bound is possible only for the price of stability (see Section 17.2.2). Second, the price of stability has a natural interpretation in many network games—if we envision the outcome as being initially designed by a central authority for subsequent use by selfish players, then the best equilibrium is an obvious solution to propose. Indeed, in many networking applications, it is not the case that agents are completely independent; rather, they interact with an underlying protocol that essentially proposes a collective solution to all participants, who can either accept it or defect from it. The price of stability measures the benefit of such protocols. Because of this interpretation, the price of stability is typically studied only for equilibrium concepts that involve no randomization, such as pure-strategy Nash equilibria. For example, since a mixed-strategy Nash equilibrium might randomize only over outcomes that are not (pure-strategy) Nash equilibria, it is not clear how to interpret it as a single proposed outcome for future use by selfish players.

The price of stability thus quantifies the necessary degradation in solution quality caused by imposing the game-theoretic constraint of stability. The goal of seeking a good equilibrium is reminiscent of the general motives of mechanism design (Part II) – designing a game outcome that (approximately) optimizes a social objective function and is also consistent with self-interested behavior.

In this book, we will only quantify the inefficiency of the worst or the best equilibrium of a game. A third interesting approach is to analyze a “typical” equilibrium. Such “average-case analyses” are notoriously difficult to define in a meaningful and analytically tractable way, however, and this approach has not yet been used successfully to study the inefficiency of equilibria.

17.2 Fundamental Network Examples

Even in very simple games, equilibria can be arbitrarily inefficient. For example, consider the Prisoner’s Dilemma, and let the players’ costs in the Nash equilibrium tend to infinity. For every reasonable objective function, the objective function value of the unique Nash equilibrium is arbitrarily larger than that of the optimal outcome.

Since the inefficiency of equilibria cannot be bounded in general, a natural goal is to identify classes of games in which equilibria are guaranteed to be approximately optimal. Fortunately, this is the case for a wide array of fundamental network models.

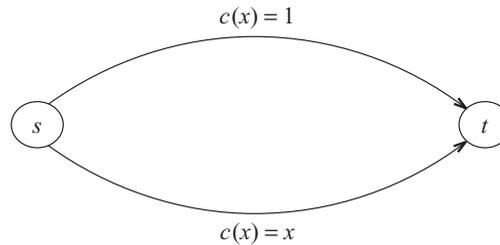


Figure 17.1. Pigou's example. The cost function $c(x)$ describes the cost incurred by users of an edge, as a function of the amount of traffic routed on the edge.

This section illustrates the concepts defined in Section 17.1 by informally introducing four such models. Chapters 18–21 study these and related models in greater depth.

17.2.1 Selfish Routing

We begin with a model of “selfish routing” that is discussed extensively in Chapter 18. We introduce the model via *Pigou's example*, which was first discussed in 1920 by the economist Pigou.

Example 17.1 (Pigou's Example) Consider the simple network shown in Figure 17.1. Two disjoint edges connect a source vertex s to a destination vertex t . Each edge is labeled with a *cost function* $c(\cdot)$, which describes the cost (e.g., travel time) incurred by users of the edge, as a function of the amount of traffic routed on the edge. The upper edge has the constant cost function $c(x) = 1$, and thus represents a route that is relatively long but immune to congestion. The cost of the lower edge, which is governed by the function $c(x) = x$, increases as the edge gets more congested. In particular, the lower edge is cheaper than the upper edge if and only if less than one unit of traffic uses it. We are interested in the price of anarchy of this game.

Suppose that there is one unit of traffic, representing a very large population of players, and that each player chooses independently between the two routes from s to t . Assuming that each player aims to minimize its cost, the lower route is a dominant strategy. In the unique equilibrium, all players follow this strategy, and all of them incur one unit of cost.

To define the optimal outcome, we assume that the objective function is to minimize the average cost incurred by players. In the above equilibrium, this average cost is 1. A simple calculation shows that splitting the traffic equally between the two links is the optimal outcome. In this outcome, half of the traffic (on the upper link) incurs cost 1, while the other half (on the lower link) experiences only $1/2$ units of cost. Since the average cost of traffic in this optimal outcome is $3/4$, both the price of anarchy and the price of stability in this game equal the ratio $1/(3/4) = 4/3$.

General selfish routing games are conceptually similar to Pigou's example, but are more complex in several respects: the network can be an arbitrarily large directed graph, different players can have different source and destination vertices, and edge cost functions can be arbitrary nonnegative, continuous, and nondecreasing functions.

One property of Pigou's example that holds more generally is that the price of anarchy and the price of stability are equal – that is, the average cost incurred by traffic is the same in all equilibria of the game. Chapter 18 proves this “essential uniqueness” property using a powerful and flexible technique called the *potential function method*. Roughly, a potential function for a game is a real-valued function, defined on the set of possible outcomes of the game, such that the equilibria of the game are precisely the local optima of the potential function. Not all games admit natural potential functions, but most of the ones discussed in this part of the book do. As we will see in Chapters 18 and 19, when a game admits a potential function, there are typically consequences for the existence, uniqueness, and inefficiency of equilibria.

One of the goals of Chapter 18 is to understand how the price of anarchy of a selfish routing game depends on different properties of the network. For example, recall that the price of anarchy in Pigou's example is precisely $4/3$. Does this bound degrade as the network size grows? As the number of distinct source and destination vertices increases? As the edge cost functions become increasingly nonlinear? If players control a nonnegligible fraction of the overall traffic? Chapter 18 provides answers to all of these questions. For example, in every network with affine cost functions (of the form $ax + b$), no matter how large and complex, the price of anarchy is at most $4/3$. With arbitrary cost functions, even with the simple network structure shown in Figure 17.1, the price of anarchy can be arbitrarily large (Exercise 17.1).

17.2.2 Network Design and Formation Games

Chapter 19 studies a diverse set of models of network formation and network design with selfish players. Here we discuss only one, with the goal of illustrating the differences between the price of anarchy and the price of stability.

We define a *Shapley network design game* as follows. Like selfish routing games, such a network design game occurs in a directed graph G . Each edge e of the graph has a fixed, nonnegative cost c_e . There are k players, and each player i is associated with a source vertex s_i and a destination vertex t_i . Player i wants to establish connectivity from its source to its destination, and its strategies are therefore the s_i - t_i paths of G . Given a choice of a path P_i by each player i , we assume that the formed network is simply the union $\cup_i P_i$ of these. The cost of this network is the sum $\sum_{e \in \cup_i P_i} c_e$ of the costs of these edges, and we assume that this cost is passed on to the players in a natural way: the cost of each edge of the formed network is shared equally by the players who use it. More formally, each player i incurs cost c_e/f_e for each edge e of its path P_i , where f_e denotes the number of players selecting paths that contain the edge e . This defines a finite noncooperative game, and we are interested in the inefficiency of the pure-strategy Nash equilibria of this game. We assume that the social objective is to minimize the cost of the formed network.

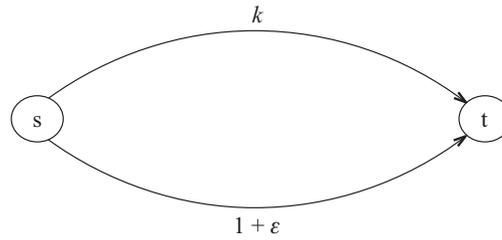


Figure 17.2. Multiple Nash equilibria in Shapley network design games (Example 17.2).

Example 17.2 Consider the network shown in Figure 17.2. There are k players, each with the same source s and destination t . The edge costs are k and $1 + \epsilon$, where $\epsilon > 0$ is arbitrarily small. In the optimal outcome, all players choose the lower edge. This outcome is also a Nash equilibrium. On the other hand, suppose that all of the players choose the upper edge. Each player then incurs cost 1, and if a player deviates to the lower edge, it pays the larger cost of $1 + \epsilon$. This outcome is thus a second Nash equilibrium, and it has cost k .

The price of anarchy of the game in Example 17.2 is roughly the number of players, and we view this as unacceptably large. This example motivates the study of the price of stability of Shapley network design games. Recall from Section 17.1.3 that the price of stability has a natural interpretation in network formation games – it measures the inefficiency of the network that a designer would propose to selfish players (i.e., the best equilibrium).

The price of stability in Example 17.2 is 1. The next example shows that this is not always the case.

Example 17.3 (\mathcal{H}_k example) Consider the network shown in Figure 17.3. There are k players, all with the same sink t , and $\epsilon > 0$ is arbitrarily small. For each $i \in \{1, 2, \dots, k\}$, the edge (s_i, t) has cost $1/i$. In the optimal outcome, each player i chooses the path $s_i \rightarrow v \rightarrow t$ and the cost of the formed network

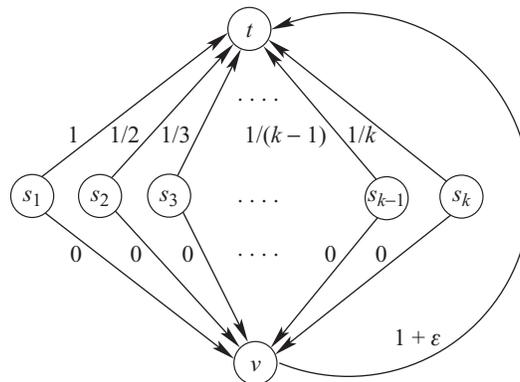


Figure 17.3. The price of stability in Shapley network design games can be at least \mathcal{H}_k (Example 17.3).

is $1 + \epsilon$. This is not a Nash equilibrium, as player k can decrease its cost from $(1 + \epsilon)/k$ to $1/k$ by switching to the direct path $s_k \rightarrow t$ – indeed, this direct path is a dominant strategy for the k th player. Arguing inductively about the players $k - 1, k - 2, \dots, 1$ shows that the unique Nash equilibrium is the outcome in which each player chooses its direct path to the sink. The cost of this outcome is exactly the k th harmonic number $\mathcal{H}_k = \sum_{i=1}^k (1/i)$, which is roughly $\ln k$. The price of stability can therefore be (arbitrarily close to) \mathcal{H}_k in Shapley network design games.

Our emphasis on pure-strategy Nash equilibria and Example 17.3 motivate the following two questions.

- (1) Does every Shapley network design game possess at least one pure-strategy Nash equilibrium? (Recall from Example 17.7 that not all games have such equilibria.)
- (2) What is the largest-possible price of stability in Shapley network design games?

Chapter 19 uses the potential function method discussed in Section 17.2.1 to resolve both of these questions. This method answers the first question in the affirmative, and also shows that the price of stability in every k -player Shapley network design game is at most \mathcal{H}_k . In other words, for each value of k , the game in Example 17.3 has the largest-possible price of stability.

Chapter 19 also discusses the price of anarchy and stability in other models of selfish network design and formation.

17.2.3 Scheduling Games

Our next example is a load-balancing scenario, where the goal is to spread several identical “jobs” evenly across a number of identical “machines.” This is a very simple type of *scheduling* problem; this and much more general scheduling models have been extensively studied and have numerous applications (see Chapter 20). We focus on this special case to illustrate a nonutilitarian objective function, mixed-strategy Nash equilibria, and the interaction between the two.

Concretely, we assume that there are m jobs and m machines for some integer $m \geq 1$. Players correspond to jobs. The strategy set of each player is the set of machines. Each player i seeks to minimize the total number of players (including i itself) that select its machine. This defines a noncooperative game. The pure-strategy Nash equilibria of this game are precisely the $m!$ outcomes in which each player selects a distinct machine. There are additional mixed-strategy Nash equilibria, as we discuss below.

To study the price of anarchy, we require an objective function. Thus far, we have studied only utilitarian objective functions, where the goal is to minimize the sum of the players’ costs. Here, motivated by the goal of load-balancing, we focus primarily on the egalitarian objective of minimizing the number of jobs on the most crowded machine. This objective is called *makespan minimization* in the scheduling literature. The set of optimal outcomes under this objective coincides with the set of pure-strategy Nash equilibria, and these outcomes all have makespan equal to 1.

In the previous two examples, we studied only pure-strategy equilibria, where the objective function value of an equilibrium is clear. In the present application, we

also consider mixed-strategy Nash equilibria. Such an equilibrium naturally induces a probability distribution on the set of game outcomes. Specifically, since we assume that the random choices made by distinct players are independent, the probability of a given strategy profile is the product of the probabilities that each player selects its prescribed strategy.

We define the objective function value of a mixed-strategy Nash equilibrium as the expectation, over this induced distribution on the game outcomes, of the objective function value of an outcome. Thus the objective function value of a mixed-strategy Nash equilibrium is its “expected objective function value.” As we now show, allowing mixed-strategy Nash equilibria can increase the price of anarchy in scheduling games.

Example 17.4 (Balls and Bins) In the above example with m jobs and m machines, suppose that every player selects a machine uniformly at random. We claim that this defines a mixed-strategy Nash equilibrium. To prove it, adopt the first player’s viewpoint. Since each of the other $m - 1$ players chooses a machine uniformly at random, all m machines appear equally loaded. More formally, let X_{ij} denote the indicator random variable for the event that player i selects the machine j . If the first player selects machine j , then it incurs a cost of $1 + \sum_{i>1} X_{ij}$. By linearity of expectation, its expected cost on this machine is $1 + \sum_{i>1} \mathbf{E}[X_{ij}] = 2 - 1/m$. Since this expected cost is independent of the machine j , every pure strategy of the first player is a best response to the mixed strategies chosen by the other players. As a consequence, every mixed strategy of the first player is also a best response (recall Section 17.3.4). This argument clearly applies to the other players as well, and hence this set of mixed strategies forms a Nash equilibrium.

What is the objective function value of this mixed-strategy Nash equilibrium – the expected value of the most heavily loaded machine? We emphasize that this expectation $\mathbf{E}[\max_j \{\sum_i X_{ij}\}]$ is not the same as the maximum expected load, $\max_j \{\mathbf{E}[\sum_i X_{ij}]\}$, which is only 1. Intuitively, the expected number of jobs on the most crowded machine is governed by the severity of the “collisions” that occur when the players select machines in a randomized and uncoordinated way. This nontrivial problem, typically called the *balls into bins* problem, is classical and has been thoroughly analyzed. In particular, the objective function value of the above mixed-strategy Nash equilibrium is $\Theta(\log m / \log \log m)$ as m grows large. (See Chapter 20.)

Collisions due to independent random choices therefore give rise to significant inefficiency: the price of anarchy with respect to pure-strategy Nash equilibria in this example is 1 for every $m \geq 1$, whereas the price of anarchy with respect to mixed-strategy Nash equilibria is $\Omega(\log m / \log \log m)$ as m grows large.

Example 17.4 shows that the price of anarchy can depend fundamentally on the choice of equilibrium concept; recall the fourth question of Section 17.1.2. As an aside, we note that it also illustrates the dependence of the price of anarchy on the choice of objective function. Specifically, consider the utilitarian objective function, where the goal is to minimize the sum of the players’ costs. The optimal outcomes again coincide with the pure-strategy Nash equilibria, and all of these have objective function

value equal to m . The objective function value of the mixed-strategy Nash equilibrium in Example 17.4 is defined as the expected sum of the players' costs, which by linearity of expectation is the same as the sum of the players' expected costs. The calculation in Example 17.4 shows that each player's expected cost equals $2 - 1/m$, and thus the objective function value of this mixed-strategy Nash equilibrium is precisely $2m - 1$. This is in fact the worst equilibrium of the game (Exercise 17.3), and hence the price of anarchy for the utilitarian objective in this example is only $2 - 1/m$.

We could also consider the objective of minimizing the maximum expected load, instead of the expected maximum load, experienced by a player. Both of these objectives can be viewed as egalitarian objectives, and they assign the same objective function value to every pure strategy profile. In particular, these objective functions have identical optimal values. However, they typically assign different values to a profile of mixed strategies. For example, the maximum expected load experienced by a player in the mixed-strategy Nash equilibrium in Example 17.4 is only $2 - 1/m$. This is the worst equilibrium (as in Exercise 17.3), and the price of anarchy with respect to the maximum expected load of a player is therefore only $2 - 1/m$. An arguably undesirable feature of this objective is that the price of anarchy is small even though, with high probability, the players' random strategy selections produce a pure strategy profile with objective function value $\Omega(\log m / \log \log m)$ times that of optimal.

Returning to the makespan minimization objective considered in Example 17.4, Chapter 20 proves that the price of anarchy is bounded above by $O(\log m / \log \log m)$ in load-balancing games with n jobs and m machines, even when the machines are "nonuniform" in a certain precise sense. Chapter 20 also studies the price of anarchy in several variants of this scheduling game.

17.2.4 Resource Allocation Games

We next study a game that is induced by a natural protocol for allocating resources to players with heterogeneous utility functions. Chapter 21 studies such games in much greater depth.

We consider a single divisible resource – the bandwidth of a single network link, say – to be allocated to a finite number $n > 1$ of competing players. We assume that each player i has a concave, strictly increasing, and continuously differentiable utility function U_i . A *resource allocation game* is defined by the n utility functions U_1, \dots, U_n and the link capacity $C > 0$. An outcome of such a game is a nonnegative allocation vector (x_1, \dots, x_n) with $\sum_i x_i = C$, where x_i denotes the amount of bandwidth allocated to player i . We study the utilitarian objective, and are thus interested in maximizing the sum $\sum_i U_i(x_i)$ of the players' utilities.

The *proportional sharing* protocol allocates bandwidth as follows. Each user expresses its interest in receiving bandwidth by submitting a nonnegative bid b_i . The protocol then allocates all of the bandwidth in proportion to the bids, so that each user i receives

$$x_i = \frac{b_i}{\sum_{j=1}^n b_j} \cdot C \quad (17.1)$$

units of bandwidth. Player i is then charged its bid b_i . See Section 17.3 and Chapter 21 for a discussion of alternative protocols that have a similar flavor.

We assume that player payoffs are quasilinear in the sense of Section 9.3. In other words, the payoff Q_i to a player i is defined as its utility for the bandwidth it receives, minus the price that it has to pay:

$$Q_i(b_1, \dots, b_n) = U_i(x_i) - b_i = U_i\left(\frac{b_i}{\sum_{j=1}^n b_j} \cdot C\right) - b_i. \quad (17.2)$$

Assume that if all players bid zero, then all users receive zero payoff. Our restrictions on the utility function U_i ensure that the payoff function Q_i is continuously differentiable and strictly concave in the bid b_i for every fixed vector b_{-i} with at least one strictly positive component (Exercise 17.4). (As usual, b_{-i} denotes the vector of bids of players other than i .)

An *equilibrium* is a bid vector in which every user bids optimally, given the bids of the other users.

Definition 17.5 A bid vector (b_1, \dots, b_n) is an *equilibrium* of the resource allocation game (U_1, \dots, U_n, C) if for every user $i \in \{1, 2, \dots, n\}$,

$$Q_i(b_i, b_{-i}) = \sup_{\tilde{b}_i \geq 0} Q_i(\tilde{b}_i, b_{-i}). \quad (17.3)$$

The potential function method also applies to resource allocation games. This method can be used to show that, for every resource allocation game, every equilibrium bid vector induces the same allocation. Thus, every equilibrium has equal objective function value. The next example shows that equilibria in resource allocation games can be inefficient.

Example 17.6 Consider a resource allocation game in which the capacity C is 1, the first user has the utility function $U_1(x_1) = 2x_1$, and the other $n - 1$ users have the utility function $U_i(x_i) = x_i$. In the optimal allocation, the first player receives all of the bandwidth and the corresponding objective function value is 2. This allocation does not, however, arise from an equilibrium. To see why, observe that (17.1) implies that the only bid vectors that induce this allocation are those in which only the first player submits a positive bid. Such a bid vector cannot be an equilibrium, as the first player can bid a smaller positive amount and continue to receive all of the bandwidth. (See also Exercise 17.5.)

A similar argument holds whenever the first player's bid is a sufficiently large fraction of the sum of the players' bids: if the first player lowers its bid, its allocation diminishes, but the effective "price per unit of bandwidth" that it pays decreases by a large enough amount to increase its overall payoff. More formally, suppose that (b_1, \dots, b_n) is an equilibrium, and let B denote the sum of the bids. By Exercise 17.5, at least two of the bids are strictly positive. By definition, the bid b_1 satisfies (17.3). Since the payoff function Q_1 is continuously differentiable and strictly concave in the bid b_1 with b_{-1} fixed (Exercise 17.4), we can compute b_1 by differentiating the right-hand side of (17.3) and setting this

derivative to zero. Starting from the defining equation (17.2) of the function Q_i , using that $U_1(x_1) = 2x_1$ and $C = 1$, and calculating, we obtain the condition $2(B - b_1)/B^2 = 1$. For a player $i > 1$, the same calculation yields the condition $(B - b_i)/B^2 = 1$. Subtracting the second equation from the first implies that $2b_1 - b_i = B$ for every $i = 2, 3, \dots, n$. Adding these $n - 1$ equations together gives $2(n - 1)b_1 - (B - b_1) = (n - 1)B$; solving, we find that the first player's bid is only an $n/(2n - 1)$ fraction of the sum of the bids: $b_1 = nB/(2n - 1)$. In the resulting allocation, the first player obtains only an $n/(2n - 1)$ fraction of the bandwidth. As n grows large, roughly half of the bandwidth is allocated to the first player, while the rest is split equally between the other $n - 1$ players. The objective function value of this allocation is roughly $3/2$, which is only a $3/4$ fraction of the value of the optimal allocation.

Intuitively, inefficiency arises in Example 17.6 because of “market power” – the fact that a single player receives the lion's share of the total bandwidth in the optimal allocation. Indeed, resource allocation games were initially studied under the assumption that no users have nontrivial market power; in this case, equilibria are fully efficient and the price of anarchy is 1. Details are discussed in Chapter 21. Chapter 21 also uses the price of anarchy as a criterion for mechanism and protocol design; we foreshadow this work in the next section.

17.3 Inefficiency of Equilibria as a Design Metric

17.3.1 Motivation

In the previous section, we studied four natural network examples. The game was given and immutable in all of these examples, and the only question involved the quality of its equilibria. While most work on the inefficiency of equilibria has been of this form, the flexibility of the framework presented in Section 17.1.2 begs a more general question: how can we *design* a game, or modify an existing game, to minimize the inefficiency of its equilibria? This question is especially crucial in settings where equilibria are unacceptably inefficient, but directly imposing an optimal solution is impractical.

Example questions of this type include the following. Among a given class of mechanisms, which one induces a game with the best price of anarchy? Quantitatively, what is this best-possible price of anarchy? Given a game and a restricted set of options for influencing its equilibria, which option improves the price of anarchy by the maximum-possible amount? How large is the improvement? Using the measures of inefficiency described in Section 17.1.2, we can rigorously compare the performance of different solutions, and quantify the efficiency loss obtained by an optimal solution.

These goals are conceptually the same as those of algorithmic mechanism design, studied in Part II of this book. However, much of the work we describe below and in the notes (Section 17.4) differs from the bulk of the material in Part II in three technical respects. First, while Part II largely concerns the design of strategyproof mechanisms in which truthful revelation is a dominant strategy for every player, we

study the equilibria of mechanisms that are not generally strategyproof. For example, in the proportional sharing mechanism described in Section 17.2.4, the strategy space of a player (nonnegative bids) does not coincide with its type space (utility functions), and no player has a dominant strategy. These differences are typically motivated by practical considerations, as we discuss in Section 17.3.2. Second, some of the research described in Section 17.4 considers games without private preferences. In these cases, the design problem is nontrivial because the mechanism designer lacks full control over the allocation of resources. Optimally influencing traffic in a selfish routing network by pricing the network edges is one example of such a problem. Third, in much of the work discussed in Section 17.4, the problem is not to design a good mechanism from scratch, but rather to leverage a limited amount of power to improve the equilibria of a given game as much as possible.

17.3.2 An Example: The Proportional Sharing Mechanism

We now informally describe one example of how the inefficiency of equilibria can serve as a design metric. Chapter 21 discusses the following result in greater detail, and Section 17.4 discusses additional examples.

Recall the resource allocation games of Section 17.2.4, where n players compete for a divisible link with capacity C . We studied the proportional sharing mechanism, in which each player i submits a bid b_i to the mechanism, the mechanism allocates all of the bandwidth to the players in proportion to their bids, and every player then pays its bid. This mechanism induces a noncooperative game; as proved in Chapter 21, the price of anarchy of every such game is at least $3/4$. We next strive to surpass this efficiency guarantee and ask: how can we modify the mechanism so that the price of anarchy is always even closer to 1?

The answer to this question depends crucially on the class of mechanisms that we are willing to consider. If we impose no restrictions on the allowable mechanisms, then a version of the VCG mechanism (see Chapter 9) always induces a game for which the price of anarchy equals 1. However, this solution is “more complicated” than the proportional sharing mechanism in two ways. First, the communication from the players to the mechanism is more involved; each player must submit a representation of its entire utility function, as opposed to a single bid. Second, the communication from the mechanism back to the players is also more complicated in the following sense. In the proportional sharing mechanism, allocations can be completely summarized by the bids and a single additional parameter, the *price* of bandwidth. To see this, consider a bid vector (b_1, \dots, b_n) for a link with capacity C . Set a price p equal to B/C , where B is the sum of the bids. The (proportional) allocation to each player i is then simply its bid b_i divided by this price. While the allocations of the VCG mechanism can be similarly interpreted in terms of prices, different players are generally allocated bandwidth according to different prices.

The simplicity of the proportional sharing mechanism – that the communication both to and from the mechanism is limited – makes it particularly suitable for implementation in large communication networks. Is there a mechanism that retains these appealing properties and has strictly smaller worst-case efficiency loss? Chapter 21 shows that the answer is “no” – for an appropriate definition of “bounded

communication,” every equally simple mechanism can induce a game that has a price of anarchy of at most $3/4$. The proportional sharing mechanism is therefore optimal among all mechanisms meeting natural, desirable implementation constraints.

17.4 Notes

The observation that self-interested behavior can lead to a socially inefficient outcome is an old one; see, for example, Dubey (1986), Rapoport and Chammah (1965), and the references therein. The idea of quantifying the inefficiency of equilibria using an objective function and an approximation measure is much newer. The concept of the price of anarchy originated in Koutsoupias and Papadimitriou (1999), where it was called the *coordination ratio*. Koutsoupias and Papadimitriou studied a generalization of the scheduling games described in Section 17.2.3. Papadimitriou (2001) introduced the term “the price of anarchy.” The price of stability was first studied in Schulz and Stier Moses (2003); the terminology is from Anshelevich et al. (2004). Several earlier works, and in particular Mason (1985), anticipated these concepts. See also Satterthwaite and Williams (1989) and Moulin and Shenker (2001), who studied additive notions of efficiency loss in mechanism design applications.

The measures of inefficiency discussed in Section 17.1 are similar to and motivated by several well-established concepts in theoretical computer science. One example is the *approximation ratio* of a heuristic for a (typically NP-hard) optimization problem, defined as the worst ratio between the objective function value of the solution produced by the heuristic and that of an optimal solution (Vazirani, 2001). While the approximation ratio measures the worst-case loss in solution quality due to insufficient computational effort, the price of anarchy measures the worst-case loss arising from insufficient ability (or willingness) to control and coordinate the actions of selfish individuals.

The novel notions of “equilibrium outcomes” alluded to in Section 17.1.2 are described in Mirrokni and Vetta (2004) and Goemans et al. (2005). Tennenholtz (2002) also proposed relaxing the assumption that players reach a Nash equilibrium, and examining the consequences for the players’ payoffs. The inefficiency of other equilibrium concepts has also been studied; see work by Christodoulou and Koutsoupias (2005) on correlated equilibria (Section 17.5), Andelman et al. (2007) on strong Nash equilibria (Section 17.6), and Hayrapetyan et al. (2006) on equilibria in the presence of coalitions of players.

Pigou’s example (Example 17.1) is from Pigou (1920). Selfish routing networks and their equilibria were defined formally by Wardrop (1952) and Beckmann et al. (1956). The potential function method originates in Beckmann et al. (1956) and was later developed by Rosenthal (1973), Monderer and Shapley (1996), Roughgarden and Tardos (2002), and Anshelevich et al. (2004). Shapley network design games were first studied by Anshelevich et al. (2004), and Example 17.3 is from Anshelevich et al. (2004). Example 17.2 was given in an earlier paper by Anshelevich et al. (2003). The scheduling games of Section 17.2.3 and Example 17.4 are due to Koutsoupias and Papadimitriou (1999). See Motwani and Raghavan (1995) for a discussion of the balls into bins problem. The proportional sharing mechanism is due to Kelly (1997), and

Example 17.6 is from Johari and Tsitsiklis (2004). For further references on the four network models of Section 17.2, see Chapters 18–21.

The results mentioned in Section 17.3.2 are from Johari and Tsitsiklis (2006), and are discussed in further detail in Chapter 21. Chapter 21 also covers variants of the VCG mechanism in which users submit only a single bid, rather than an entire utility function. These mechanisms are not (and can not) be single-price in the sense of Section 17.3.2, however.

We conclude these notes with examples of how measures of inefficiency have been used to compare different mechanisms and different strategies for influencing equilibria in the network models explored in Section 17.2. Several approaches to improving the equilibria of a selfish routing network have been considered, including pricing the network edges, and routing a small fraction of the traffic in a centralized manner. The goal is then to leverage the limited amount of design power to minimize the price of anarchy. For details on this literature, see Roughgarden (2005, Chapters 5–6) and the references therein.

Motivated by the network design games of Section 17.2.2 and Example 17.3, Chen et al. (2007) studied how to design cost-sharing methods to minimize the inefficiency of equilibria in the resulting network game. One of the contributions in Chen et al. (2007) is an analogue of the result described in Section 17.3.2 for resource allocation mechanisms: among all cost-sharing methods that are “oblivious” to the network structure in a certain precise sense, the Shapley cost-sharing method of Section 17.2.2 minimizes the worst-case price of stability. On the other hand, cost-sharing methods that can leverage information about the network topology can outperform Shapley cost-sharing methods (Chen et al., 2007).

Finally, for the scheduling games of Section 17.2.3, Christodoulou et al. (2004) and Immorlica et al. (2005) design machine scheduling policies to improve the inefficiency of equilibria. Informally, such a policy can be used to prioritize one player over another, thereby causing different players to incur different costs on a common machine. As shown in Christodoulou (2004) and Immorlica (2005), even very simple scheduling policies reduce the price of anarchy from logarithmic in the number of machines (Example 17.4) to a small constant.

Bibliography

- E. Anshelevich, A. Dasgupta, J. Kleinberg, É. Tardos, T. Wexler, and T. Roughgarden. The price of stability for network design with fair cost allocation. In *Proceedings of the 45th Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 295–304, 2004.
- E. Anshelevich, A. Dasgupta, É. Tardos, and T. Wexler. Near-optimal network design with selfish agents. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing (STOC)*, pp. 511–520, 2003.
- N. Andelman, M. Feldman, and Y. Mansour. Strong price of anarchy. In *Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2007. In press.
- M. J. Beckmann, C. B. McGuire, and C. B. Winsten. *Studies in the Economics of Transportation*. Yale University Press, 1956.
- G. Christodoulou and E. Koutsoupias. On the price of anarchy and stability of correlated equilibria of linear congestion games. In *Proceedings of the 13th Annual European Symposium on Algorithms (ESA)*, pp. 59–70, 2005.

- G. Christodoulou, E. Koutsoupias, and A. Nanavati. Coordination mechanisms. In *Proceedings of the 31st Annual International Colloquium on Automata, Languages, and Programming (ICALP)*, volume 3142 of *Lecture Notes in Computer Science*, pp. 345–357, 2004.
- H. Chen, T. Roughgarden, and G. Valiant. Designing networks with good equilibria. Manuscript, 2007.
- P. Dubey. Inefficiency of Nash equilibria. *Mathematics of Operations Research*, 11(1):1–8, 1986.
- M. X. Goemans, V. S. Mirrokni, and A. Vetta. Sink equilibria and convergence. In *Proceedings of the 46th Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 142–151, 2005.
- A. Hayrapetyan, É. Tardos, and T. Wexler. The effect of collusion in congestion games. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing (STOC)*, pp. 89–98, 2006.
- N. Immorlica, L. Li, V. S. Mirrokni, and A. S. Schulz. Coordination mechanisms for selfish scheduling. In *Proceedings of the First Annual International Workshop on Internet and Network Economics (WINE)*, volume 3828 of *Lecture Notes in Computer Science*, pp. 55–69, 2005.
- R. Johari and J. N. Tsitsiklis. Efficiency loss in a network resource allocation game. *Mathematics of Operations Research*, 29(3):407–435, 2004.
- R. Johari and J. N. Tsitsiklis. Characterization theorems for smooth market-clearing mechanisms. Submitted, 2006.
- F. P. Kelly. Charging and rate control for elastic traffic. *European Transactions on Telecommunications*, 8(1):33–37, 1997.
- E. Koutsoupias and C. H. Papadimitriou. Worst-case equilibria. In *Proceedings of the 16th Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 1563 of *Lecture Notes in Computer Science*, pp. 404–413, 1999.
- L. G. Mason. Equilibrium flows, routing patterns and algorithms for store-and-forward networks. *Large Scale Systems*, 8:187–209, 1985.
- R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.
- D. Monderer and L. S. Shapley. Potential games. *Games and Economic Behavior*, 14(1):124–143, 1996.
- H. Moulin and S. Shenker. Strategyproof sharing of submodular costs: Budget balance versus efficiency. *Economic Theory*, 18(3):511–533, 2001.
- V. S. Mirrokni and A. Vetta. Convergence issues in competitive games. In *Proceedings of the 7th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX)*, pp. 183–194, 2004.
- C. H. Papadimitriou. Algorithms, games, and the Internet. In *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing (STOC)*, pp. 749–753, 2001.
- A. C. Pigou. *The Economics of Welfare*. Macmillan, 1920.
- A. Rapoport and A. M. Chammah. *Prisoner's Dilemma*. University of Michigan Press, 1965.
- R. W. Rosenthal. A class of games possessing pure-strategy Nash equilibria. *International Journal of Game Theory*, 2(1):65–67, 1973.
- T. Roughgarden. *Selfish Routing and the Price of Anarchy*. MIT Press, 2005.
- T. Roughgarden and É. Tardos. How bad is selfish routing? *Journal of the ACM*, 49(2):236–259, 2002.
- A. S. Schulz and N. Stier Moses. On the performance of user equilibria in traffic networks. In *Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 86–87, 2003.
- M. A. Satterthwaite and S. R. Williams. The rate of convergence to efficiency in the buyer's bid double auction as the market becomes large. *Review of Economic Studies*, 56(4):477–498, 1989.
- M. Tennenholtz. Competitive safety analysis: Robust decision-making in multi-agent systems. *Journal of Artificial Intelligence Research*, 17:363–378, 2002.
- V. V. Vazirani. *Approximation Algorithms*. Springer, 2001.

EXERCISES

457

J. G. Wardrop. Some theoretical aspects of road traffic research. In *Proceedings of the Institute of Civil Engineers, Pt. II*, volume 1, pp. 325–378, 1952.

Exercises

- 17.1** Suppose that we modify Pigou's example (Example 17.1) so that the lower edge has the cost function $c(x) = x^d$ for some $d \geq 1$. What is the price of anarchy of the resulting selfish routing network, as a function of d ?
- 17.2** Suppose we modify the \mathcal{H}_k example (Example 17.3) so that all of the network edges are undirected. In other words, each player i can choose a path from s_i to t that traverses each edge in either direction. What is the price of stability in the resulting Shapley network design game?
- 17.3** Recall the scheduling game in Example 17.4, with m players and m machines. Prove that the price of anarchy of this game with respect to the utilitarian objective function is precisely $2 - 1/m$.
- 17.4** Let U_i be a concave, strictly increasing, and continuously differentiable univariate function. Define the function Q_i as in (17.2). Prove that Q_i is continuously differentiable and strictly concave in b_i for every fixed nonnegative vector b_{-i} with at least one strictly positive component.
- 17.5** Prove that every equilibrium of a resource allocation game (Definition 17.5) has at least two strictly positive components.

